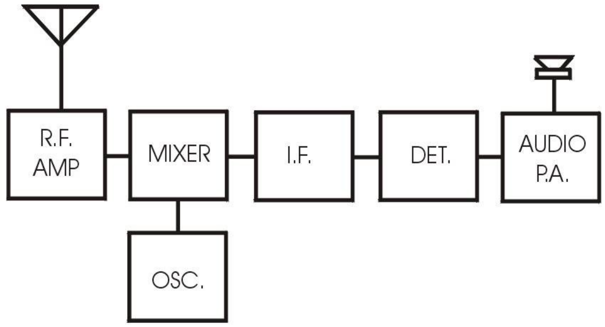
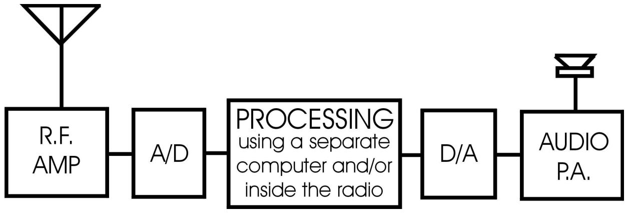
SDR 101 (A Software Defined Radio Primer)

by David Kennett W8KFJ

January 2015



Old Fashioned Superheterodyne Radio


R.F. AMP
A/D
PROCESSING using a separate computer and/or inside the radio
D/A
AUDIO P.A.

Newfangled SDR (Software Defined Radio)

Things are changing! Out with the old, in with the new.
Better radios for less money will do it.

My introduction to Software Defined Radios

About ten years ago I attended the Dayton Hamvention with my buddy Art, W8IKN (now a silent key). Art suggested I attend a forum with him on the subject of Software Defined Radios, by Gerald Youngblood of Flex-Radio. I said something like, "I don't need no stinkin' computer to control my radio." He suggested that it might be more than that, and that it might be interesting. Well, I did go with him, and now my perception of how radios are built is forever changed. Thank you Art!

He immediately bought a Flex SDR1000 , but I had just upgraded to an Icom IC-706MKIIG, so my wallet remained tightly closed. I finally weakened a few years later when the Flex 3000 was introduced. What a revelation! The panadapter opened a wonderful new world that let me know what was happening in the frequencies around me. Those digital filters are razor-sharp, and the software features continue to improve and expand. I should have done this sooner!

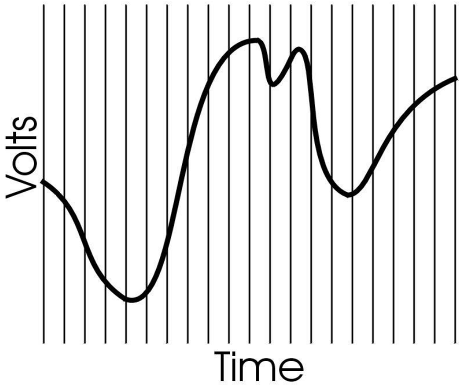
Later, I built a $129 Peaberry SDR kit, used a $900 Hermes board from Apache Labs (developed by the Open HPSDR group) to make an SDR, sold the Flex 3000, and recently added a $1280 Elad FDM-DUO transceiver. I am constantly amazed by the continuing improvement in performance, and new features that have been enabled by this technology. But as I listen on the air and talk to people, I can't help but feel there's still a lack of understanding of what's really different about Software Defined Radios. So where do we begin?

DSP (Digital Signal Processing)

To understand what makes an SDR, we first have to know a little about Digital Signal Processing. It's a pretty straightforward process. A signal (any analog signal, such as sound, or even the whole HF radio spectrum) is digitized by an Analog to Digital (A/D) converter. The resulting data is then processed with software written by those living in a special world beyond the understanding of us mere mortals. Then the processed data is converted back to an analog waveform by a Digital to Analog (D/A) converter. In the case of a receiver, it's converted to analog sound so we can hear it.

Analog to Digital conversion

Each vertical line represents a point in time when a sample of the waveform voltage is taken. The more data that is used for each sample, and the more often the sample is taken, the more precise the data.



For a direct sampling receiver, it is common to sample over 120 million times per second, and use 16 bits for each sample, allowing over 65 thousand possible values for each sample.

The more often a signal is sampled, and the more bits of data used for each sample, the better the data can represent the original waveform. Normally a signal must be sampled at least twice the highest frequency to be digitized. If you wanted to digitize 100 KHz, then you must sample at least 200 Khz (creating 200 kilosamples per second), 60 MHz to sample a 30 MHz signal, etc. Most SDRs use 16 bits (two bytes) for each sample.

A Software Defined Radio could be considered DSP (Digital Signal Processing) on steroids. Somebody's early definition of SDR went something like this: “The radio must have at least *some* of its usual analog components **replaced** by software”. I suppose one could make the case that *any* DSP in a radio would replace at least a couple of analog components. That way, just about any radio made in the last ten years or so is an SDR. Since the term SDR is pretty broad, how do we know whether we're buying a marginal SDR or the real McCoy? Read on!

So What's the difference?

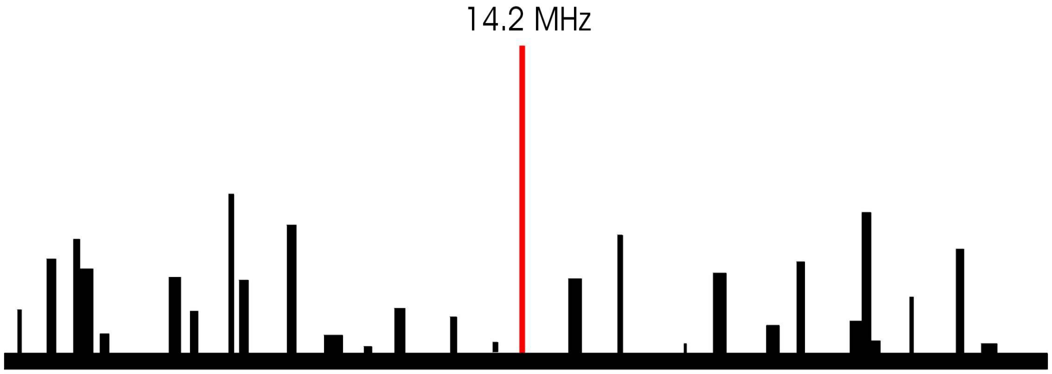
DSP (Digital Signal Processing) functions have been used for *noise reduction, automatic notch filters,* and *noise blankers* for years. However, by **my** more specific definition, at least the IF filters must be digital to qualify. If the radio manufacturer mentions that you can buy additional filters (2.3KHz, 500 Hz, 6KHz, etc), then I say there's simply not enough DSP. Software filters just work so much better, and they're free! You can have any variety you want. Getting rid of those nasty analog components improves performance, while the cost of digital keeps dropping. The best analog components are maybe one percent precision, and they change with age. The digital equivalents can be carried out to many decimal places mathematically, and they <u>never</u> change or drift.

“Sound Card” SDRs

The first mainstream SDR was the Flex SDR 1000 (*circa* 2003), and it used the A/D converters in the sound cards present in many computers of the day. The “sound cards” might have been actual plug-in cards, or were included on the mother board, or could be a complete separate plug-in USB (Universal Serial Bus) module. But the name “Sound Card SDR” stuck. Those “sound cards” could sample at 48 KHz or 96 KHz, and some of the best could even sample at 192 KHz. However, many built-in sound modules were of marginal quality, and sound modules made by the likes of Turtle Beach and M-Audio proved superior. Incidentally, I'll be talking about receivers, but the digital process for transmitting exists, and it's pretty much the inverse of the receive process.

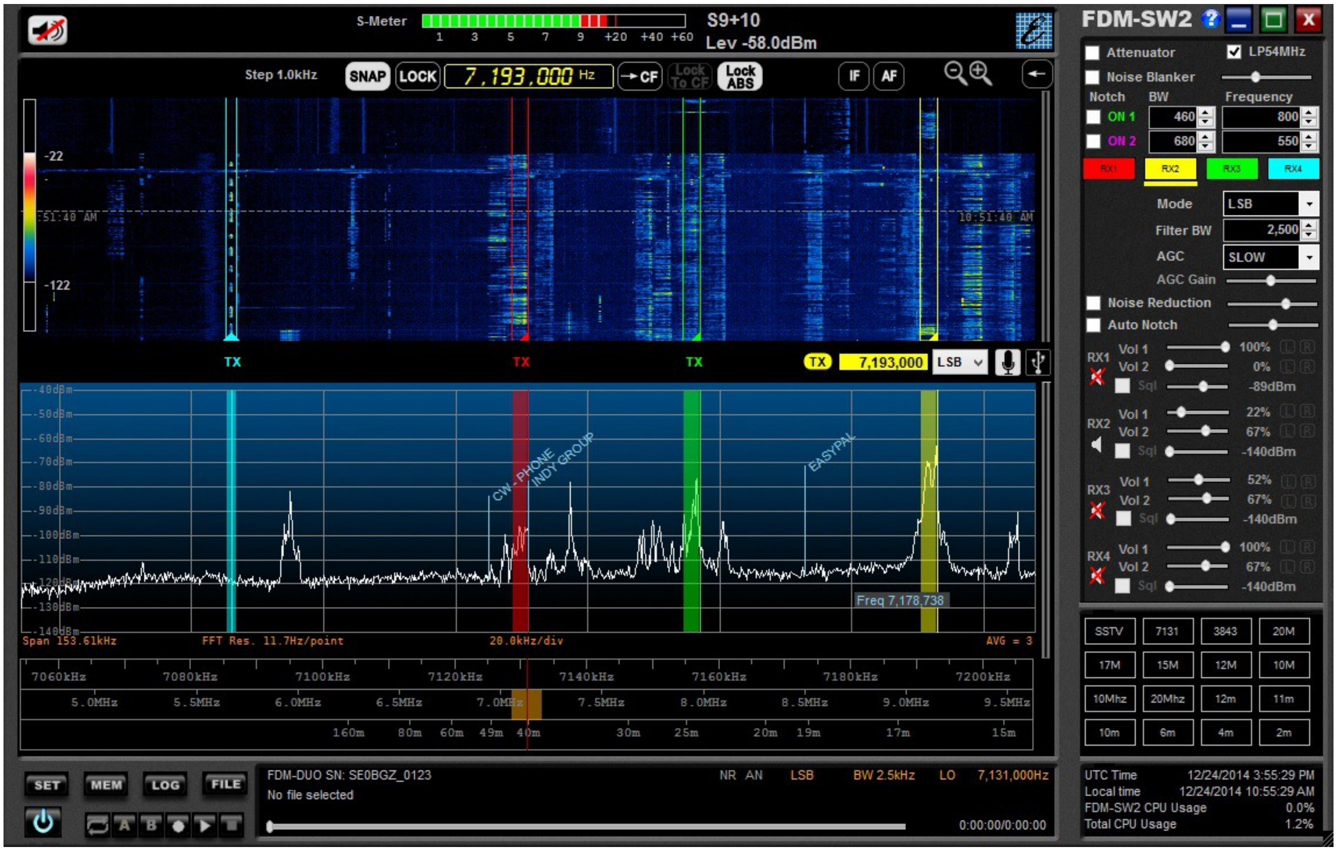
Here's where it gets a little tricky. In a four-part article in QEX (Jul/Aug 2002, Sep/Oct 2002, Nov/Dec 2002, and Mar/Apr 2003), Gerald Youngblood described a process that created I and Q signals that could then be used to digitize a small slice of the radio spectrum – limited by the sampling rate of the sound card. It took a four part article to describe the process, but for now, just recognize that two signals (called I – for In-phase, and Q – for Quadrature, or 90 degrees phase shifted from the I signal) were created, digitized, and made available to the wizards of DSP math.

As an example, a 14.2 MHz variable frequency oscillator
is mixed with R F from the antenna.



48 KHz, or 96 KHz, or 192 KHz of spectrum
centered around 14.2 MHz is digitized,
depending on sampling rate

A local VFO (Variable Frequency Oscillator) is mixed with the received signal, and through the magic of I and Q, a spectrum is created, centered around the VFO frequency. The I and Q signals are then digitized using the left and right stereo channels of the “sound card.” From here on, it's pure digital processing in the computer, until it's time to convert back to analog to feed speakers or headphones. This is the technique used by the Flex SDR 1000. and later, the A/D and D/A converters were included in the Flex 5000, 3000, and 1500, so no sound card was required. Using this technique, a number of inexpensive SDR kits appeared, like the Softrock kits from Tony Parks, the Peaberry kit from AE9RB, and others. They used the free open-source Power SDR software used by Flex, as well as other free software like HDSDR, SDR#, CuSDR, and much more.



Mode LSB
Filter BW 2,500
AGC SLOW

FDM-DUO SN: SE0BGZ_0123
No file selected

UTC Time 12/24/2014 3:55:29 PM
Local time 12/24/2014 10:55:29 AM
FDM-SW2 CPU Usage 0.0%
Total CPU Usage 1.2%

This is a screen capture of the SW2 software included with the Elad FDM-DUO. An S meter and frequency readout are obvious at the top. Below that, a "waterfall" shows the recent history of what's been happening in the spectrum display, which fills a large portion of the screen below that. Although it really does more than just display a frequency spectrum, since you can tune the radio, and make many adjustments as well. This interactive frequency display is often called a “panadapter”. The three horizontal bars below that allow you to drag the displayed spectrum anywhere. The top bar moves the spectrum at the speed you drag it. The middle bar drags it somewhat faster (the range of the top bar is the orange block in the center of the middle bar). The bottom bar can instantly tune the entire range of the radio. This “three-tier” technique is unique in an SDR to Elad. There are four frequencies presently tuned, represented by the blue, red, green, and yellow vertical bars. The yellow bar is now active, and is the selected transmit frequency, using the microphone input on lower sideband, at 7,193,000 Hz. Several SSB stations can be seen, along with a strong carrier to the right of the red bar. The location of any memories within the shown spectrum, along with their diagonal labels can be seen. In the same fashion, I can elect to show labels from a variety of sources, such as DX clusters, or even call letters from the CW decoding program *CW skimmer*. There are various adjustments at the right, and below that are 16 quick memory buttons. A more complete memory function is available at the MEM button in the lower left. In the lower right corner is a block showing a variety of information. To the left is another block with more information, including the recorder status. In the lower left corner are the recorder controls, along with buttons to access memories, a built-in logging feature, a setup menu, and a “POWER” button. The recorder can record just the audio, or the entire visible spectrum – even if it's 6 MHz wide. That file can then be played, and any stations in that spectrum can be tuned as if it were live. Most SDR software is similar, although features vary.

As a side benefit, a panadapter (spectrum display) could be created without adding hardware, and is included on the computer screen of pretty much every SDR. These are usually very high resolution, and display what's happening in real time around your operating frequency. Using software, it's even possible to listen to multiple signals displayed in the panadapter. There's a whole world out there of inexpensive hardware and free software for HF, VHF, and UHF. It's a great way to have a top-notch radio for cheap!

This digital processing was a major departure from the components used in a typical superheterodyne receiver. No more IF mixer, IF filters, detectors, AGC amps, equalizers, VOX circuits, etc. Best of all, no more alignment! No more pots to get scratchy or components to drift. The mathematics is either perfect or (occasionally) disastrous failure.

DDC/DUC (Digital Down Conversion/Digital Up Conversion) is much better

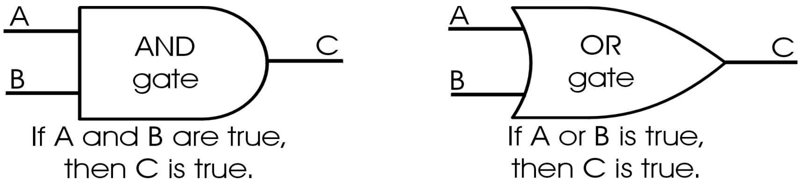
There was still one weakness, however. I and Q were created as analog signals. And for this technique to work well, I and Q must be perfectly matched. They must be exactly the same amplitude, and exactly 90 degrees apart. This is hard to maintain in the analog world, but easy (perfect) in the digital domain. For this to happen, the signal must be digitized **before** it is converted down to low frequencies usually needed by the sound card. It is common to sample at a high enough rate to digitize all frequencies below 60 MHz. This includes 160 through 6 meters, along with everything in between, the AM broadcast band, and the low frequency band down to 10 KHz or so. That's a lot of data!

Once the whole spectrum is digital, I and Q are simply created mathematically, so a sound card will no longer be needed to digitize the received signal. In fact, several sets of I and Q can be created, each with its own panadapter on entirely different frequencies. You can have one panadapter for 80 meters, and maybe another for 20 meters. Some radios allow even more. In turn, each panadapter can have the capability to tune multiple stations. For example, my new FDM-DUO can have two individual panadapters, each as wide as 384 KHz, with the capability to tune up to four frequencies in each pandapter. If I want only one panadapter, it can be as wide as 6 MHz. I can listen to all eight frequencies if I wish, but generally I watch the activity on the panadapter to judge band activity, or look for open spots for a quick QSY. Keep in mind, this is all with **one** actual receiver. Some radios have **two** actual receivers, so multiply everything I just said by two.

Digitizing at those higher frequencies has its cost though. And I mean dollar cost! High speed A/D converters are more expensive. And what do we do with all that data? A PC can't begin to handle it! Remember, the A/D converter must sample at least twice the highest frequency to be digitized. So for 10 meters, that's 60 MHz, or 120 MHz for 6 meters. Sampling at 122.88 MHz has become popular, and it works for 6 meters. Even higher frequencies are used in some radios. But sampling at 100 MegaHertz creates a thousand times the data created by sampling at 100 KiloHertz. That problem is solved by another expensive item, the FPGA.

So, what's an FPGA?

The letters stand for Field Programmable Gate Array, and just like it says, this array of gates may be programmed to be “wired” together inside the chip any way you wish. And yes, we're talking about those simple AND, NAND, OR, and NOR gates. They are the building blocks of all things digital, and an FPGA may have millions of them.



The desired configuration of all these gates is loaded into the FPGA from memory at start-up, and the FPGA then becomes whatever you want it to be. Keep in mind that this can all be changed by replacing the start-up memory (firmware) at any time.

Think about it! By changing the hardware configuration in the FPGA, the radio can be morphed into something totally different years after it was manufactured. By contrast, a microprocessor can be programmed to **do** just about anything, but it is always a microprocessor. It's even possible, and sometimes useful, to program a part of the FPGA to **be** a microprocessor!

So why would we want to do something in an FPGA instead of a microprocessor? Signal processing in a microprocessor takes clock cycles to perform each step. And many times one step must be completed before the next can be started. (The answer to one step might be needed as input to the next step.) With even the fastest processor, these steps can add up. In an FPGA, the delay need only be the very minimal propagation delay through the gates.

In a real-world example, the HPSDR wizards have programmed all of the CW operation into the FPGA of the Hermes board (and other radios from Apache Labs), virtually eliminating delay in a critical area, (especially for those 50 wpm guys wanting full break-in.) The best delay figure I have heard using microprocessors is about 8 ms. This pretty much eliminates the biggest complaint about SDRs – delay, or latency. Only the limited amount of data necessary for the sound we are listening to, and enough data to display the panadapter are passed on to the computer, or maybe to a microprocessor inside the radio.

Hermes board. The big square black chip in the center is the FPGA.
The only adjustment on the board is a trimmer capacitor at the top,
so no more alignment!

A Timeline of Radio Technology

1900? to 1928? In the beginning days of radio, there were crystal sets, Tuned RF, and direct conversion.

1918 to ? The invention of the superheterodyne receiver improved sensitivity, and most importantly, selectivity. It's interesting to note that it took about ten years for the superhet to take over. At first, there were few radio stations, so who needed selectivity? That means just about every radio made in the last 100 years or so is a superhet.

1995 First digital cell phone (yet another world).

2003 to ? Software defined radios. I'm going to separate these into categories, hopefully to improve understanding. In my mind , if it's a superhet, it can't be an SDR. You may, of course, categorize them any way you wish.

Type 1: "sound card" SDR. A maximum digitized bandwidth of 192 KHz is typical. It's relatively inexpensive, and provides improved performance over the superhet. A benefit is that more than one signal can be tuned at the same time within the digitized bandpass, and the ubiquitous panadapter is usually interactive, making tuning a breeze. Examples are: Flex-Radio 5000, 3000, and 1500, as well as inexpensive kits like Softrock and Peaberry. The Elecraft KX3 is interesting in that basically falls into this category, but has knobs and does not require a computer. There is no panadapter included, and only one signal can be tuned at a time.

Type 2: DDC/DUC, or direct sampling. The RF is directly sampled at the antenna, at a frequency at least twice the highest frequency to be tuned. Since everything is thus digitized, the I and Q are created mathematically from all that data. The result is a nearly perfect I and Q. A frequency of 122.88 MHz is common for receiving through the six meter band. By creating I and Q digitally rather than analog, the intermod products and dynamic range are greatly improved. Multiple panadapters can be created at various frequencies and of different widths anywhere in the tunable spectrum, adding flexibility beyond our wildest dreams, with the capability to tune many stations (even in different bands) simultaneously. As a type 2 SDR, it still needs a separate computer. The higher cost of high speed A/D converters and FPGAs has limited the practicality of this technique, although the cost has been dropping rapidly. A few years ago an FPGA could cost thousands of dollars. The one used in my new FDM DUO sells for $53 at Digi-Key. Examples of type 2 SDRs are the Hermes board, and other offerings from Apache Labs.

Type 2a: This is just like type 2, except all the DSP is done in the radio, leaving just control and display of the panadapter to the computer. Examples are the 6000 Series from Flex-Radio.

Type 2b: This is just like type 2a, except that knobs, buttons, and a display have been added, and a computer is not even required. There are only a few radios in this category, such as the SunSDRMB1 from Expert Electronics, the Adat ADT200A, and the Elad FDM-DUO, which puts the radio, along with three microprocessors and a small LCD display inside a very small enclosure. This combination is surprisingly capable and inexpensive. It's only five watts, but an amplifier can solve that. Add a computer, and you can tune eight stations at once. You can have many panadapters – probably more than you can fit on your computer screen(s).

How do we know what were getting?

Check the sampling rate! If it's around 100 MHz, it's direct sampling. If it's around 100 Khz, it's a “sound card” type. Ask what kind of FPGA is used. A DDC/DUC (direct sampling) radio sporting an FPGA is incredibly flexible, and can deliver top notch performance at a price that continues to drop. If there's any mention of roofing filters, or double or triple conversion, it's a superhet.

Some seem to think that the Icom 7700 is an SDR. But Icom's spec sheet specifically says that it's a superhet! You can stuff a superhet with DSP, but it's still a superhet. You can replace a buggy with an automobile, but you have to remove the horses to let the automobile “do it's stuff”. The superheterodyne receiver is the “horse-and-buggy” of the 21st century.

What does this mean?

Well, in the near future – or maybe even now, I see direct sampling (DDC/DUC) SDRs delivering better quality at a lower price than superhets. With a higher price and poorer performance, it will not be practical to continue making superheterodyne radios. We shall see! There is much more to learn about SDRs, and things keep changing every day. My hope is that you now have a sufficient foundation to pursue this fascinating world. Just don't bother making any more superhets for me.

w8kfj@yahoo.com

Some SDR Links

http://en.wikipedia.org/wiki/List_of_software-defined_radios
http://www.rtl-sdr.com/roundup-software-defined-radios/
https://sdrzone.com/
http://www.scoop.it/t/low-cost-software-defined-radio-sdr-panorama?page=1
https://apache-labs.com/1001/Ham-Radio-Products.html
http://www.flexradio.com/
http://ecom.eladit.com/
http://www.nparc.org/2014/Presentations2014/SDR_SOTAandSOTF.pdf
http://www.hdsdr.de/index.html
http://arvideonews.com/hrn/